

СИСТЕМА ВИЗУАЛИЗАЦИИ ДИНАМИЧЕСКИХ МОДЕЛЕЙ

Д. И. Мазовка

ВВЕДЕНИЕ

Моделирование реального мира является актуальной на сегодняшний день проблемой. Она возникает во многих сферах деятельности человека. Решение сводится к созданию параметрической модели, учитывающей специфику конкретной задачи. Однако процесс построения модели является сложным из-за необходимости учета множества факторов, присутствующих в реальном прототипе.

В то же время, имея только параметрическую модель, наглядно сложно оценить промежуточные и конечные результаты моделирования. Для решения этой проблемы используют визуализацию – представление модели реального мира в виде двумерного изображения.

В данной работе был рассмотрен класс задач визуализации сцен реального мира, достаточно независимых от наполнения и сложности, и построен универсальный алгоритм для их решения.

1. АНАЛИЗ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ

Введем базовые понятия, необходимые для анализа задачи.

Объектом задачи моделирования является окружающий реальный мир, определим его через понятие *среда*. Замкнутую часть среды, моделирование которой не учитывает влияние извне, назовем *сценой*. Тогда *модель сцены* – формальное описание сцены, объектов и процессов в ней. В дальнейшем под термином «сцена» будем подразумевать именно это понятие.

Будем рассматривать сцену, как множество *объектов* – моделей физических объектов реального мира. Объекты определяются конечным набором своих *свойств* – описаний характеристик реальных прототипов.

Рассматривая типы и взаимозависимости свойств, введем разбиение множества объектов на классы: статические и динамические, простые и сложные объекты, геометрические объекты, источники освещения и камеры. Классификация является необходимой для дальнейшего решения задачи.

Динамичность среды ведет к динамичности модели сцены, ее объектов и их свойств. То есть идет процесс изменение сцены во времени. Для каждого момента времени нам нужно получить *кадр* – фиксацию свойств сцены в определенный момент времени. На практике для программной реализации алгоритмов получения кадров используют графические биб-

лиотеки, организующие доступ к аппаратному обеспечению, называемые *рендером* [1].

Для работы рендер предоставляет пользователю набор инструкций и типов данных [1]. *Стоимостью инструкции* назовем время выполнения инструкции рендера.

Таким образом, задачу можно сформулировать следующим образом: дана модель сцены, необходимо произвести такое отображение свойств ее объектов в типы данных рендера и реализовать такой алгоритм построения последовательности инструкций рендера, чтобы время генерации отдельного кадра в последовательности кадров было меньше $1/30$ секунды.

2. ОБЩАЯ СХЕМА АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

Решение поставленной задачи представляется в виде композиции решений подзадач. Разбиение на подзадачи происходит на основе анализа взаимодействия различных классов объектов. В процессе решения подзадач формируются так называемые *группы отображения*, которые представляют собой наборы из одного простого геометрического объекта и нуля или более источников освещения. Набор групп отображения назовем *очередью отрисовки*. В результате композиции решений подзадач мы получаем некоторую очередь отрисовки, которая преобразуется в последовательность инструкций рендера оптимальным образом.

В итоге общая схема алгоритма решения задачи визуализации может быть представлена в виде последовательности этапов:

1. Этап подготовки данных.
2. Этап моделирования процессов в сцене.
3. Этап подготовки визуализации.
 - 3.1. Формирование групп отображения.
 - 3.2. Формирование очередей отрисовки.
4. Этап визуализации.
 - 4.1. Формирование последовательности инструкций рендера.
 - 4.2. Вызов инструкций рендера.
5. Переход к шагу 2.

3. ПРАКТИЧЕСКОЕ РЕШЕНИЕ И ПРИМЕР ИСПОЛЬЗОВАНИЯ

В основе реализации решения задачи визуализации лежит представленный выше общий алгоритм. Практическое решение представляет собой программную систему, получающую на вход сцену и генерирующую последовательность кадров на выходе.

Система состоит из набора компонент, определяющих этапы обработки и необходимые типы данных. Основными компонентами являются:

base (определение множества классов, задающих каркас системы), scene (работа на уровне сцены, ее объектов и их свойств), render (связь системы с библиотекой рендера). Вспомогательные компоненты: application, resource, entry point, constructor и serializers. Все компоненты были сделаны в виде динамически подключаемых библиотек.

Разработанная система ориентирована прежде всего на пользователя персонального компьютера. Поэтому WindowsXP была выбрана в качестве целевой платформы, из-за ее распространенности и доступности необходимых системных библиотек [2]. В качестве рендера была выбрана одна из широко используемых библиотек – DirectX9.

Основным требованием к системе является ее быстродействие. Поэтому было необходимо использовать язык, позволяющий организовать наиболее тесный контакт системы с оборудованием и операционной системой, добившись тем самым наилучшей скорости работы. В то же время этот язык обязан был быть высокоуровневым, поддерживающим объектно-ориентированное программирование. Поэтому для реализации нашей системы был выбран язык программирования C++.

Методология работы с системой состоит из нескольких этапов:

1. Определение возможности использования системы для решения задачи.
2. Подготовка входных данных.
3. Ввод данных в систему.
4. Анализ результата.

Пример визуализации сцены показан на Рис.1, реальный прототип изображен на Рис.2.

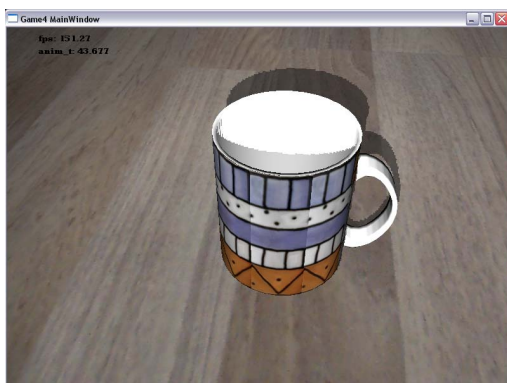


Рис. 1



Рис. 2

ЗАКЛЮЧЕНИЕ

В работе рассмотрена задача визуализации динамических сцен. В результате теоретического исследования было показано, каким образом реальный мир может быть смоделирован путем определения таких понятий

как сцена, объекты, свойства объектов. Были проанализированы свойства объектов и, учитывая сходную природу реальных прототипов, было дано соответствующее разбиение объектов на классы. В итоге это позволило сформулировать задачу визуализации для динамических сцен в общем виде и решить ее как в теоретическом, так и в практическом смысле. В итоге была разработана программная система.

Область применения системы достаточно широка: от создания презентаций, до использования в интерактивных приложениях, ориентированных на использование трехмерной графики. Система решает поставленную задачу визуализации в общем виде, что отличает ее от множества других, специализированных систем. То есть, если специализированные системы визуализации работают с объектами, то наша система работает с типами, классами объектов, позволяя таким образом абстрагироваться от частных задач и дать возможность решать какие-то их подмножества в целом. Это значит, что использование разработанной системы избавляет от необходимости создания специализированных алгоритмов визуализации при решении различных задач.

Литература

1. Microsoft DirectX9 SDK documentation
2. *Джеффри Рихтер*, Windows для профессионалов, Мн. 2003

О СТАТИСТИЧЕСКОМ АНАЛИЗЕ ЦЕПИ МАРКОВА ПЕРЕМЕННОЙ ДЛИНЫ

М.В. Мальцев

1. ОСНОВНЫЕ ПОНЯТИЯ

Цепи Маркова высокого порядка широко применяются для анализа стационарных процессов [1-4]. Существенным недостатком данной модели является экспоненциальное возрастание количества параметров при увеличении порядка. В связи с чем, была разработана модель цепи Маркова переменной длины.

Пусть $A = \{0, 1, \dots, N-1\}$ – пространство состояний мощности $2 \leq N \leq \infty$; $x_i^j = (x_i, x_{i+1}, \dots, x_j)$ – строка длительности $|x_i^j| = j - i + 1$ ($i < j$); $uw = (u_1, u_2, \dots, u_{|u|}, w_1, w_2, \dots, w_{|w|})$ – конкатенация строк u, w ; $(X_t \in A)_{t \in \mathbb{G}}$ – стационарная цепь Маркова s -го порядка с вероятностями одношаговых переходов